EXHIBIT G

Exhibit C-23

<u>Invalidity of U.S. Patent No. 7,069,560 ("'560 Patent") by Finin II</u>

As shown in the claim chart below, the asserted claims of the '560 patent are invalid under 35 U.S.C. §§ 102(a) and/or (b)[1] as anticipated by "KQML as an Agent Communication Language" by Tim Finin, et al. ("Finin II"), and/or are invalid under 35 U.S.C. § 103 as obvious in view of Finin II, or in combination with the reference(s) specifically identified in the following claim chart or one or more other references identified in Defendants' Preliminary Invalidity Contentions, Defendants' First Supplemental Invalidity Contentions, and Defendants' Second Supplemental Invalidity Contentions (collectively "Defendants' Invalidity Contentions").

The additional KIF/KQML documents relied upon herein include, but are not limited to:

- Yannis Labrou, et al., "Semantics for an Agent Communication Language" (1997)[2] ("Labrou"). Labrou provides further descriptions and examples of the types of syntax and expressions possible with the KIF/KQML architecture that is described in Finin II.
- Tim Finin, et al., Specification of the KQML Agent-Communication Language (1993)[3] ("KQML 1993").
- Yannis Labrou, et al., A proposal for a new KQML Specification (1997)[4] ("KQML 1997")

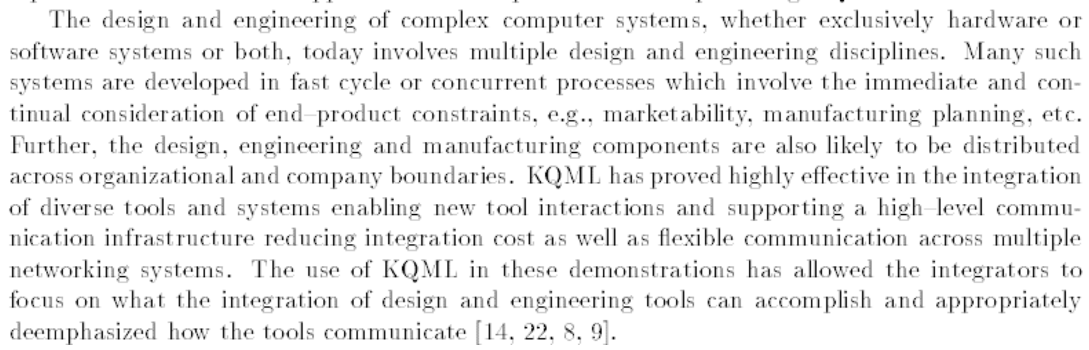
To the extent a finder of fact determines that the references cited herein do not teach certain limitations in the asserted claims, such limitations would have been inherent and/or obvious. These claims are also invalid as obvious in view of Finin II alone or in combination with other prior art references, including, but not limited, to the prior art identified in the Cover Pleading of Defendants' Invalidity Contentions, the prior art described in the claim charts attached in Appendix C, and/or the prior art identified in Appendix D.

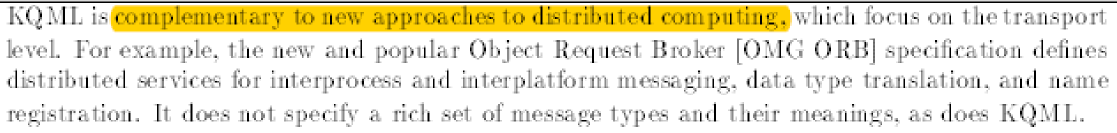
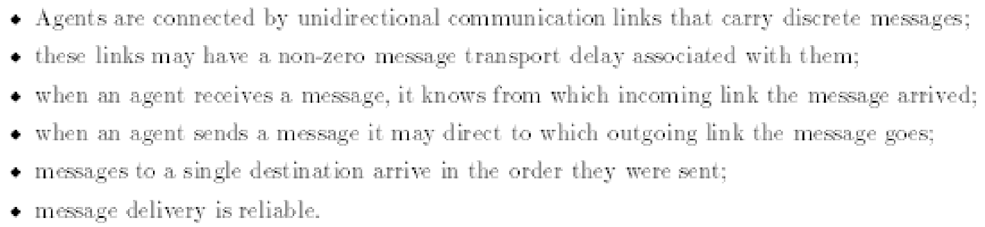
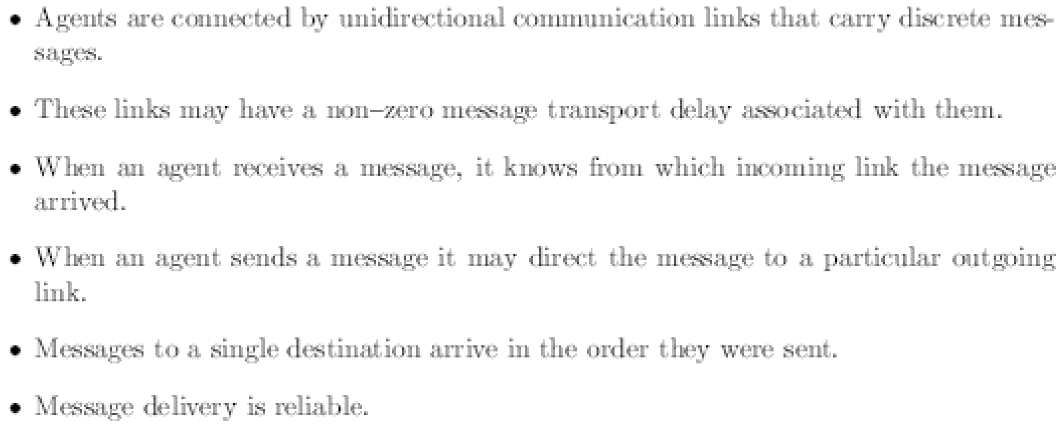
Defendants' Invalidity Contentions are based, in part, upon Defendants' present understanding of the asserted claims and IPA's

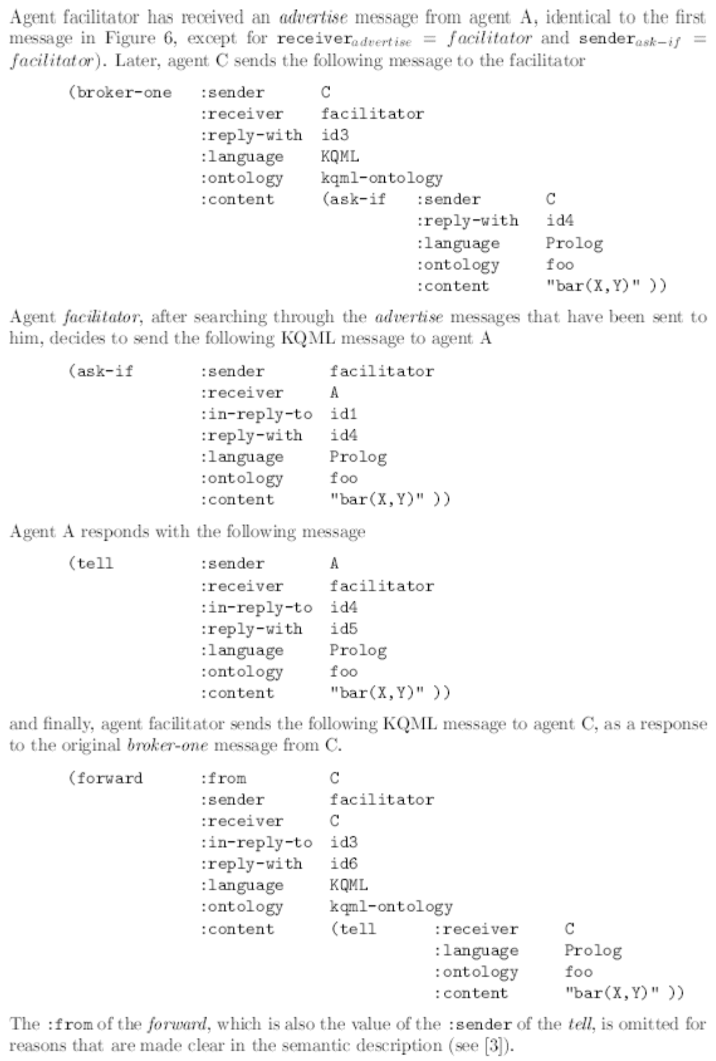
[1] Finin II was published in Jeffrey M. Bradshaw, "Software Agents," *American Association for Artificial Intelligence* (1997), before the January 5, 1999 earliest possible priority date of the '560 patent.
[2] Published in Munindar P. Singh, et al., Intelligent Agents IV Agent Theories, Architectures, and Languages, ATAL '97, vol 1365.
[3] Upon information and belief, this KQML specification was published in 1993 and is available at https://www.csee.umbc.edu/csee/research/kqml/papers/kqmlspec.pdf
[4] Upon information and belief, this KQML specification was published in 1997 and is available at https://www.csee.umbc.edu/csee/research/kqml/papers/kqml97.pdf.

apparent interpretations of the asserted claims in its July 10, 2019 Preliminary Infringement Contentions ("Infringement Contentions") and Defendants' investigation to date. Defendants are not adopting IPA's constructions or apparent constructions, nor are Defendants admitting to the accuracy of any particular contention or construction. The citations provided in the charts below are exemplary rather than exhaustive and Defendants reserve the right to rely upon additional references uncovered through further searching, other portions of the cited references and/or other portions of references cited within these Invalidity Contentions. Defendants further incorporate by reference the reservation of rights identified in the cover pleading to these Invalidity Contentions as though fully set forth herein.

	'560 Patent Claim Language	**Invalidity in View of Prior Art**
1(p)	A computer-implemented method for communication and cooperative task completion among a plurality of distributed electronic agents, comprising the acts of	To the extent the preamble is found to be limiting, Finin II discloses a computer-implemented method for communication and cooperative task completion among a plurality of distributed electronic agents, comprising the acts of claim 1. *See* Claim Chart for U.S. Patent No. 6,851,115 in view of Finin II, Ex. A-23 ("'115 chart"), claim 1.
1(a)	a plurality of service-providing electronic agents;	Finin II discloses a plurality of service-providing electronic agents. *See* '115 chart, claim 1.
1(b)	a distributed facilitator agent functionally distributed across at least two computer processes, the facilitator agent capable of bi-directional communications with the plurality of service-providing electronic agents, the facilitator agent including:	Finin II discloses a distributed facilitator agent functionally distributed across at least two computer processes, the facilitator agent capable of bi-directional communications with the plurality of service-providing electronic agents, the facilitator agent including. *See, e.g.,* Finin II at 13 (emphasis added). A facilitator is an agent that performs various useful communication services, e.g. maintaining a registry of service names, forwarding messages to named services, routing messages based on content, providing "**matchmaking**" between information providers and clients, and providing mediation and translation services. *See also,* Finin II at 17.

	’560 Patent Claim Language	Invalidity in View of Prior Art
		 The design and engineering of complex computer systems, whether exclusively hardware or software systems or both, today involves multiple design and engineering disciplines. Many such systems are developed in fast cycle or concurrent processes which involve the immediate and continual consideration of end–product constraints, e.g., marketability, manufacturing planning, etc. Further, the design, engineering and manufacturing components are also likely to be distributed across organizational and company boundaries. KQML has proved highly effective in the integration of diverse tools and systems enabling new tool interactions and supporting a high–level communication infrastructure reducing integration cost as well as flexible communication across multiple networking systems. The use of KQML in these demonstrations has allowed the integrators to focus on what the integration of design and engineering tools can accomplish and appropriately deemphasized how the tools communicate [14, 22, 8, 9]. *See also,* KQML 1993, p. 29 (emphasis added). [A]gents do not communicate directly with each other. Instead, they communicate with their local facilitators, and facilitators communicate with each other. In effect, the agents form a “federation” in which they surrender their communication autonomy to facilitators; hence the name of the architecture. Messages from agents to facilitators may be directed or undirected. Undirected messages have content but no addresses. It is the responsibility of the facilitators to route such messages to agents able to handle them. In performing this task, facilitators can **go beyond simple pattern match** – they can translate messages, they can decompose problems into subproblems, and they can schedule the work on these subproblems. In some cases, this can be done interpretively (with messages going through the facilitator); in other cases, it can be done in one-shot fashion (with the facilitator setting up specialized links between individual agents and then stepping out of the picture) *See also,* KQML 1993, p. 1 (emphasis added).

	'560 Patent Claim Language	Invalidity in View of Prior Art
		 KQML is complementary to new approaches to distributed computing, which focus on the transport level. For example, the new and popular Object Request Broker [OMG ORB] specification defines distributed services for interprocess and interplatform messaging, data type translation, and name registration. It does not specify a rich set of message types and their meanings, as does KQML. *See also,* KQML 1993, p. 6.  • Agents are connected by unidirectional communication links that carry discrete messages; • these links may have a non-zero message transport delay associated with them; • when an agent receives a message, it knows from which incoming link the message arrived; • when an agent sends a message it may direct to which outgoing link the message goes; • messages to a single destination arrive in the order they were sent; • message delivery is reliable. *See also,* KQML 1997, p. 1. The point of this point-to-point message transport abstraction is to provide a simple, uniform model of communication for the outer layers of agent-based programs. This should make agent-based programs and APIs easier to design and build. *See also,* KQML 1997, p. 1.  • Agents are connected by unidirectional communication links that carry discrete messages. • These links may have a non–zero message transport delay associated with them. • When an agent receives a message, it knows from which incoming link the message arrived. • When an agent sends a message it may direct the message to a particular outgoing link. • Messages to a single destination arrive in the order they were sent. • Message delivery is reliable.

	’560 Patent Claim Language	Invalidity in View of Prior Art
		See also, KQML 1997, p. 36 (disclosing bi-directional communication). 

Agent facilitator has received an *advertise* message from agent A, identical to the first message in Figure 6, except for $\texttt{receiver}_{advertise} = \textit{facilitator}$ and $\texttt{sender}_{ask-if} = \textit{facilitator}$). Later, agent C sends the following message to the facilitator

```
(broker-one   :sender      C
              :receiver    facilitator
              :reply-with  id3
              :language    KQML
              :ontology    kqml-ontology
              :content     (ask-if   :sender       C
                                     :reply-with   id4
                                     :language     Prolog
                                     :ontology     foo
                                     :content      "bar(X,Y)" ))
```

Agent *facilitator*, after searching through the *advertise* messages that have been sent to him, decides to send the following KQML message to agent A

```
(ask-if       :sender        facilitator
              :receiver      A
              :in-reply-to   id1
              :reply-with    id4
              :language      Prolog
              :ontology      foo
              :content       "bar(X,Y)" ))
```

Agent A responds with the following message

```
(tell         :sender        A
              :receiver      facilitator
              :in-reply-to   id4
              :reply-with    id5
              :language      Prolog
              :ontology      foo
              :content       "bar(X,Y)" ))
```

and finally, agent facilitator sends the following KQML message to agent C, as a response to the original *broker-one* message from C.

```
(forward      :from          C
              :sender        facilitator
              :receiver      C
              :in-reply-to   id3
              :reply-with    id6
              :language      KQML
              :ontology      kqml-ontology
              :content       (tell     :receiver     C
                                       :language     Prolog
                                       :ontology     foo
                                       :content      "bar(X,Y)" ))
```

The `:from` of the *forward*, which is also the value of the `:sender` of the *tell*, is omitted for reasons that are made clear in the semantic description (see [3]).

	'560 Patent Claim Language	Invalidity in View of Prior Art
		Finin II discloses this limitation as identified above. This limitation is also obvious in view of Finin II combined with the knowledge of a person having ordinary skill in the art and/or any one or more of the references identified in the corresponding limitation of Ex. C-X. As Ex. C-X shows, each of these references also discloses this limitation. A person having ordinary skill in the art would have been motivated to combine Finin II with the knowledge of a person having ordinary skill in the art and/or any one or more of the references identified in the corresponding limitation of Ex. C-X rendering this limitation obvious based on one or more of the motivations to combine identified in Section II.A.3.e of the cover pleading to Defendants' Preliminary Invalidity Contentions and because each of these references relate to the same field of software agent technology and distributed computing environments.
1(c)	an agent registry that declares capabilities for each of the plurality of service-providing electronic agents currently active within the distributed computing environment; and	Finin II discloses an agent registry that declares capabilities for each of the plurality of service-providing electronic agents currently active within the distributed computing environment. *See* '115 chart, claim 1(a).
1(d)	a facilitating engine operable to interpret a service request as a base goal, the facilitating engine further operable for generating a goal satisfaction plan associated with the base goal, wherein the goal satisfaction plan involves:	Finin II discloses a facilitating engine operable to interpret a service request as a base goal, the facilitating engine further operable for generating a goal satisfaction plan associated with the base goal, wherein the goal satisfaction plan involves. *See* '115 chart, claims 1(e), (h).
1(e)	using reasoning to determine sub-goal requests based on non-syntactic decomposition of the base goal and using said reasoning to co-ordinate and schedule efforts by the service-providing electronic agents for fulfilling the sub-goal	Finin II discloses using reasoning to determine sub-goal requests based on non-syntactic decomposition of the base goal and using said reasoning to co-ordinate and schedule efforts by the service-providing electronic agents for fulfilling the sub-goal requests in a cooperative completion of the base goal. *See* '115 chart, claim 1(g)-(i).

	’560 Patent Claim Language	**Invalidity in View of Prior Art**
	requests in a cooperative completion of the base goal; and	Finin II discloses this limitation as identified above. This limitation is also obvious in view of Finin II combined with the knowledge of a person having ordinary skill in the art and/or any one or more of the references identified in the corresponding limitation of Ex. C-X. As Ex. C-X shows, each of these references also discloses this limitation. A person having ordinary skill in the art would have been motivated to combine Finin II with one or more of the references identified in the corresponding limitation of Ex. C-X rendering this limitation obvious based on one or more of the motivations to combine identified in Section II.A.3.e of the cover pleading to Defendants’ Preliminary Invalidity Contentions and because each of these references relate to the same field of software agent technology and distributed computing environments.
1(f)	wherein the plurality of service-providing electronic agents and the distributed facilitator agent communicate using an interagent Communication Language (ICL), wherein the ICL includes:	Finin II discloses wherein the plurality of service-providing electronic agents and the distributed facilitator agent communicate using an interagent Communication Language (ICL), wherein the ICL includes. *See* ’115 chart, claims 1(a)-(c).
1(g)	a layer of conversational protocol defined by event types and parameter lists associated with one or more of the events, wherein the parameter lists further refine the one or more events.	Finin II discloses a layer of conversational protocol defined by event types and parameter lists associated with one or more of the events, wherein the parameter lists further refine the one or more events. *See* ’115 chart, claims 1(b)-(c).
20	A computer architecture as recited in claim 1 wherein the distributed facilitator agent includes a planning component executing within a first computer process and an execution component executing within a second computer process.	Finin II discloses a computer architecture as recited in claim 1 wherein the distributed facilitator agent includes a planning component executing within a first computer process and an execution component executing within a second computer process of claim 20. *See, e.g.,* Finin II at 3.